1 Usage example

```
## First, load the libraries we need
library(ape) # for tree manipulation
library(ade4) # for the dpcoa function

## Make some fake data
nOTUs = 200; nSamples = 50
tree = rtree(nOTUs)
otuTab = data.frame(matrix(rexp(nOTUs * nSamples), nrow = 200))

## make a matrix of the distances between the OTUs and correct for the
## fact that the distance is non-euclidean
patristicDist = cophenetic(tree)
patristicDist = cailliez(as.dist(patristicDist))
output = dpcoa(otuTab, patristicDist, scannf = FALSE, nf = 2)

## plot the communities...
plot(output$l2)
## and species
plot(output$l1)
```

2 Supplemental Figures

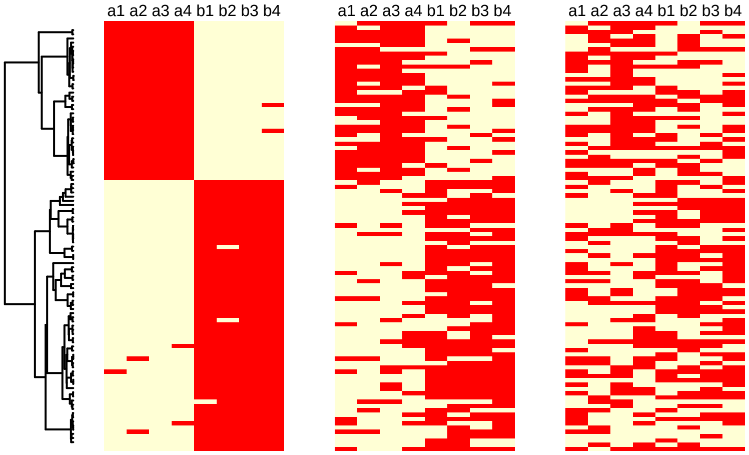

Figure 1: A representation of the three sets of simulated data. On the left we have the tree. Each rectangular block is a different simulation, with noise levels .01, .2, and .39. Each column corresponds to a location (community), and red/white corresponds to the presence/absence of a species at that location.

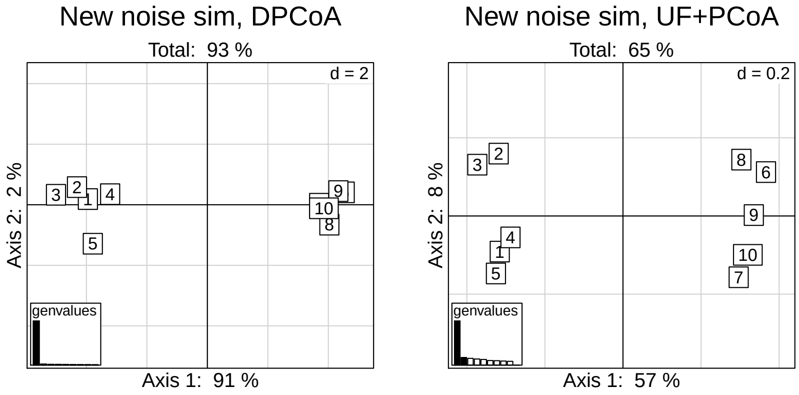

Figure 2: Noise simulation. Given a random tree with 500 leaves, about 100 were chosen at random to represent "real" OTUs. Real OTUs from one half of the tree were simulated as being present in half of the ten communities, and the other half of the OTUs were simulated as being present in the other half of the communities. The remaining OTUs were "noise", and were present in exactly one of the groups. We can see that both DPCoA and UniFrac can still differentiate between the two groups, but DPCoA suppresses the noise more than UniFrac does.

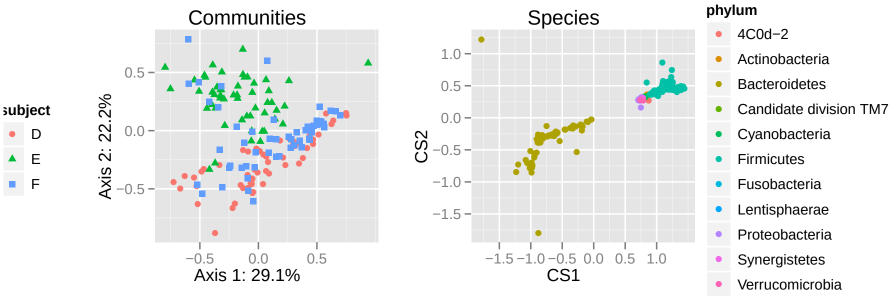

Figure 3: Community and species points for DPCoA. Note that the spaces of the two plots are identical, so we could superimpose one on the other. We have plotted them separately for better readability. We see that there are two outlying OTUs, one in the top left and one in the bottom left of the OTU plot.

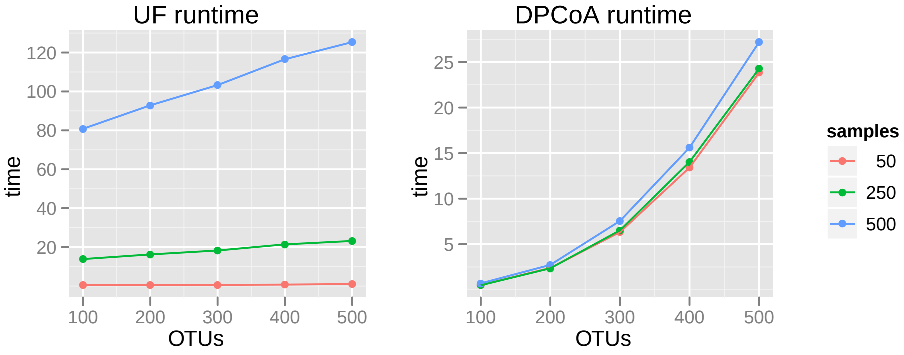

Figure 4: Runtimes of DPCoA and UniFrac. We can see that the runtime of UniFrac is linear in the number of OTUs and super-linear in the number of samples, while DPCoA is quadratic in the number of OTUs and depends very little on the number of samples. Weighted UniFrac was omitted from the analysis because there is no efficient implementation in R.